



# **ATID Application Development Framework Reference Manual –Camera**

Revision: Ver. 0.3

Date: September, 2013

ATID Co.,Ltd

## Table of Contents

Table of Contents .....	2
Acronym .....	5
Revision History .....	6
1 .NET API Reference .....	7
1.1 Enumerations .....	7
1.1.1 RESULT .....	7
1.1.2 CAM_BPP .....	7
1.1.3 CAM_COL_SP .....	8
1.1.4 CAMERAMSG .....	8
1.1.5 CAM_RESOL .....	8
1.1.6 CONT_OPTIONS .....	9
1.1.7 EFFECT .....	9
1.1.8 EV_OPTIONS .....	10
1.1.9 FLIP_MIRROR .....	11
1.1.10 IMG_ENCODE_FORMAT .....	11
1.1.11 PM_OPTIONS .....	11
1.1.12 SAT_OPTIONS .....	11
1.1.13 SEN_CTRL_CODE .....	12
1.1.14 WHITE_BAL .....	13
1.1.15 CAM_DRV .....	13
1.2 Structures .....	14
1.2.1 stBRIGHTNESS .....	14
1.2.2 stCONTRAST .....	14
1.2.3 stEXPOSURE .....	14
1.2.4 stSATURATION .....	15
1.2.5 stZOOM_CAPA .....	15
1.2.6 CAPTURE_SETUP_INFO .....	15
1.2.7 SENSOR_SETUP_INFO .....	15
1.2.8 SENSOR_CAPA .....	16
1.3 Delegates .....	17
1.3.1 CAMERACALLBACK .....	18
1.4 Methods .....	19
1.4.1 Open .....	19
1.4.2 Close .....	19
1.4.3 IsOpen .....	19
1.4.4 SetCallback .....	19
1.4.5 SetHandle .....	20

1.4.6	StartPreview .....	20
1.4.7	StopPreview .....	21
1.4.8	Capture .....	21
1.4.9	GetInfo .....	21
1.4.10	GetSensorCapa.....	22
1.4.11	SetAutoFocus .....	22
1.4.12	SetAutoFocusEx .....	22
1.4.13	SetBrightness .....	23
1.4.14	SetContrast .....	23
1.4.15	SetEffect .....	23
1.4.16	SetFlipMirror.....	24
1.4.17	SetSaturation.....	24
1.4.18	SetWhiteBalance.....	24
1.4.19	EnableFlash .....	25
2	C/C++ API Reference .....	26
2.1	Enumerations.....	26
2.1.1	CAM_RESULT .....	26
2.1.2	CAM_BPP .....	26
2.1.3	CAM_COL_SP.....	27
2.1.4	CAMERAMSG .....	27
2.1.5	CAM_RESOL.....	27
2.1.6	CONT_OPTIONS.....	27
2.1.7	EFFECT .....	28
2.1.8	EV_OPTIONS.....	28
2.1.9	FLIP_MIRROR.....	28
2.1.10	IMG_ENCODE_FORMAT .....	29
2.1.11	PM_OPTIONS .....	29
2.1.12	SAT_OPTIONS .....	29
2.1.13	SEN_CTRL_CODE.....	29
2.1.14	WHITE_BAL.....	30
2.2	Structures.....	31
2.2.1	stBRIGHTNESS .....	31
2.2.2	stCONTRAST.....	31
2.2.3	stEXPOSURE.....	31
2.2.4	stSATURATION.....	32
2.2.5	stZOOM_CAPA.....	32
2.2.6	CAPTURE_SETUP_INFO .....	32
2.2.7	SENSOR_SETUP_INFO .....	32
2.2.8	SENSOR_CAPA .....	33

2.3	Constants.....	34
2.3.1	CAM_MGS_OFFSET .....	34
2.3.2	WM_MSG_CAMAPI.....	34
2.3.3	WM_CAM_DRV_CAMERA_INITIALIZED.....	34
2.3.4	WM_CAM_DRV_CAPTURE_DONE .....	34
2.3.5	WM_CAM_DRV_CAPTURE_START_TO_SAVE .....	34
2.3.6	WM_CAM_DRV_CAPTURE_COMPLETE_SAVE .....	34
2.3.7	WM_CAM_DRV_PREVIEW_DONE .....	34
2.4	Methods.....	35
2.4.1	CamOpen.....	35
2.4.2	CamClose.....	35
2.4.3	CamIsOpen .....	35
2.4.4	CamSetHandle .....	35
2.4.5	CamStartPreview.....	36
2.4.6	CamStopPreview.....	36
2.4.7	CamCapture.....	36
2.4.8	CamGetInfo.....	37
2.4.9	CamGetSensorCapa .....	37
2.4.10	CamSetAutoFocus.....	38
2.4.11	CamSetAutoFocusEx .....	38
2.4.12	CamSetBrightness.....	38
2.4.13	CamSetContrast .....	39
2.4.14	CamSetEffect .....	39
2.4.15	CamSetFlipMirror .....	39
2.4.16	CamSetSaturation .....	40
2.4.17	CamSetWhiteBalance .....	40
2.4.18	CamEnableFlash.....	41

## Acronym

modules	descriptions
<b>AADF</b>	ATIDApplication Development Framework

## Revision History

Version	Date	Reason	Description	Author
<b>0.1</b>	2012/01/27	Draft		Y. J. CHO
<b>0.2</b>	2013/01/28	Update	- AutoFocus Mode added.	Y. J. CHO
<b>0.3</b>	2013/09/04	Update	<ul style="list-style-type: none"> <li>- the explanation of CAM_DRV added.</li> <li>- Changing enum RESULT to CAM_RESULT.</li> <li>- Changing IsOpened(Function Name) to IsOpen.</li> <li>- The explanation of enum member added.</li> <li>- typo correction.</li> </ul>	Y. J. CHO

## 1 .NET API Reference

### 1.1 Enumerations

#### 1.1.1 RESULT

The result of a call to functions.

- **RESULT\_OK**  
Function executed successfully.
- **RESULT\_INVALIDARG**  
Invalid parameter.
- **RESULT\_OUTOFMEMORY**  
Failed to assign resource.
- **RESULT\_UNSUPPORTED**  
No support command currently.
- **RESULT\_ALREADY\_OPENED**  
Cam device has opened.
- **RESULT\_ALREADY\_CAM\_SELECTED**  
Cam device was already selected.
- **RESULT\_SCAN\_SELECTED**  
2D Barcode device was already selected.  
(Camera and 2D Barcode can not be used at the same time.)
- **RESULT\_NOT\_OPENED**  
Call function without Open.
- **RESULT\_FAIL**  
Failed to perform function.
- **RESULT\_INVALID\_DEVICE**  
Not installed Cam device.

#### 1.1.2 CAM\_BPP

the image data format that is output from the Camera module

- **RawRGB8Bits**
- **RGB565**
- **YCbCr422**

### 1.1.3 CAM\_COL\_SP

Color space

- **Raw**
- **RGB**
- **YCbCr**

### 1.1.4 CAMERAMSG

the state value of the camera module transmitted through Delegate function.

- **CAMERA\_INITIALIZED**  
Initializing Camera device is complete.
- **CAPTURE**  
Not used.
- **COMPLETE**  
Saving image data in form of file is complete.
- **PREVIEW\_DONE**  
Not used.
- **START\_TO\_SAVE**  
Starting to save image data in form of file.

### 1.1.5 CAM\_RESOL

Resolution.

- **CIF**  
352 \* 288
- **QQVGA**  
160 \* 120
- **QVGA**  
320 \* 240
- **QXGA**  
2048 \* 1536
- **SVGA**  
800 \* 600
- **SXGA**  
1280 \* 1024
- **UXGA**



1600 \* 1200

- **VGA**

640 \* 480

- **XGA**

1024 \* 768

### 1.1.6 CONT\_OPTIONS

Effect of contrast

- **CONT\_0**

Not changed.

- **CONT\_N1**

Negative 1 applied.

- **CONT\_N2**

Negative 2 applied.

- **CONT\_N3**

Negative 3 applied.

- **CONT\_N4**

Negative 4 applied.

- **CONT\_N5**

Negative 5 applied.

- **CONT\_P1**

Positive 1 applied.

- **CONT\_P2**

Positive 2 applied.

- **CONT\_P3**

Positive 3 applied.

- **CONT\_P4**

Positive 4 applied.

- **CONT\_P5**

Positive 5 applied.

### 1.1.7 EFFECT

Digital effects

- **EF\_BLUE**

Blue effect.

- **EF\_GREEN**

Green effect.

- **EF\_MONO**  
Mono effect.
- **EF\_NEGATIVE**  
Negative(camera film) effect.
- **EF\_RED**  
적색효과를적용  
Red effect.
- **EF\_SEPIA**  
Sepia effect.
- **EF\_VIOLET**  
Violet effect.
- **NO\_EFFECT**  
Digital effect released.

#### **1.1.8 EV\_OPTIONS**

Exposure effect.

- **EV\_0**  
default
- **EV\_N0D3**  
- 0.3EV
- **EV\_N0D7**  
- 0.7EV
- **EV\_N1D0**  
- 1.0 EV
- **EV\_N1D3**  
- 1.3 EV
- **EV\_N1D7**  
- 1.7 EV
- **EV\_P0D3**  
0.3 EV
- **EV\_P0D7**  
0.7 EV
- **EV\_P1D0**  
1.0 EV
- **EV\_P1D3**  
1.3 EV
- **EV\_P1D7**

## 1.7 EV

### 1.1.9 FLIP\_MIRROR

Screen reverse effect.

- **FLIP**

Applying horizontal reverse effect.

- **FLIP\_AND\_MIRROR**

Releasing effect.

- **MIRR**

Applying vertical reverse effect.

- **MIRR\_FLIP\_UNSUPPORTED**

상하, 좌우반전효과적용

Applying horizontal and vertical effect.

- **NO\_MIRROR\_FLIP**

Releasing effect.

### 1.1.10 IMG\_ENCODE\_FORMAT

- **IMG\_ENCODE\_BMP**

- **IMG\_ENCODE\_EXIF**

- **IMG\_ENCODE\_GIF**

- **IMG\_ENCODE\_ICON**

- **IMG\_ENCODE\_JPG**

- **IMG\_ENCODE\_TIFF**

### 1.1.11 PM\_OPTIONS

Photometric effects.

- **PM\_NORMAL**

Normal mode

- **PM\_SPOT**

Spot mode

### 1.1.12 SAT\_OPTIONS

Saturation.

- **SAT\_025X**

0.25x

- **SAT\_05X**  
0.5x
- **SAT\_075X**  
0.75x
- **SAT\_125X**  
1.25x
- **SAT\_15X**  
1.5x
- **SAT\_175X**  
1.75x
- **SAT\_1X**  
1x(Default)
- **SAT\_2X**  
2x
- **SAT\_0X**  
0x

#### **1.1.13 SEN\_CTRL\_CODE**

Not used in .NET API

- **SEN\_CTRL\_AUTOFOCUS**  
Running Auto focus.
- **SEN\_CTRL\_BRIGHTNESS**  
Control brightness.
- **SEN\_CTRL\_CONTRAST**  
Control contrast.
- **SEN\_CTRL\_EFFECT**  
Applying Digital effect.
- **SEN\_CTRL\_EXPOSURE**  
Control exposure.
- **SEN\_CTRL\_PHOTOMETRY**  
Control photometry.
- **SEN\_CTRL\_PLIP\_MIRROR**  
Apply reverse effect.
- **SEN\_CTRL\_SATURATION**  
Control saturation.
- **SEN\_CTRL\_WHITE\_BALANCE**  
Control white balance.

#### **1.1.14 WHITE\_BAL**

White balance effect.

- **WB\_AUTO**

Auto effect.

- **WB\_CLOUDY**

Cloudy effect.

- **WB\_FLUORESCENT**

Fluorescent effect.

- **WB\_INCANDESCENT**

Incandescent effect.

- **WB\_SUNNY**

Sunny effect.

- **WB\_UNSUPPORTED**

No effect.

#### **1.1.15 CAM\_DRV**

Not used in .NET API

- **CAMERA\_INITIALIZED**

Initializing Camera device is complete.

- **CAPTURE**

Not used.

- **COMPLETE**

Saving the image data in form of a file is complete.

- **PREVIEW\_DONE**

Not used.

- **START\_TO\_SAVE**

Starting to save the image data in form of a file

## 1.2 Structures

### 1.2.1 stBRIGHTNESS

Brightness Information structure

```
Public struct stBRIGHTNESS
{
    intBrightDefault
    intBrightMax;
    intBrightMin;
    byteBrightSteps
};
```

### 1.2.2 stCONTRAST

Contrast information structure.

```
Public struct stCONTRAST
{
    intContrastDefault;
    intContrastMax;
    intContrastMin;
    byteContrastSteps
};
```

### 1.2.3 stEXPOSURE

Exposure information structure

```
Public struct stEXPOSURE
{
    intExposureDefault;
    intExposureMax;
    intExposureMin;
    byteExposureSteps
};
```

#### 1.2.4 stSATURATION

Saturation information structure

```
Public struct stSATURATION
{
    int SaturationDefault;
    int SaturationMax;
    int SaturationMin;
    byte SaturationSteps
};
```

#### 1.2.5 stZOOM\_CAPA

The structure of information of zoom function

```
Public struct stZOOM_CAPA
{
    bool bZoomSupport;
    uint MaxZoom;
    int ZoomStep;
};
```

#### 1.2.6 CAPTURE\_SETUP\_INFO

The structure of the necessary information while capture image.

```
Public struct CAPTURE_SETUP_INFO
{
    IMG_ENCODE_FORMAT imgFormat
    int nJpegQuality;
    SENSOR_SETUP_INFO sensorSetup;
    string savePath
};
```

#### 1.2.7 SENSOR\_SETUP\_INFO

Constructor which for changing the setting of the image sensor

```
Public struct SENSOR_SETUP_INFO
{
```

```

bool bAutoFocus;
bool bEnableAF;
CAM_BPP bpp;
int Brightness;
bool bSupportFlash;
CAM_COL_SP ColorSpace;
CONT_OPTIONS Contrast;
EFFECT Effect;
EV_OPTION Exposure;
FLIP_MIRROR FlipMirror;
PM_OPTIONS PhotoMetry;
CAM_RESOL Resol;
SAT_OPTIONS Saturation;
WHITE_BAL WhiteBalance;
int Zoom;

```

```
};
```

- bAutoFocus : setting auto focusing execution.  
true = run auto focusing  
false = not run auto focusing
- bEnableAF : not used.
- bpp : the image data format that is output from the Camera Module.
- Brightness : value of brightness.
- bSupportFlash : whether or not flash light supported.
- ColorSpace : color space value.
- Contrast : contrast value.
- Effect : digital effect
- Exposure : exposure value.
- FlipMirror : screen reverse mode value.
- PhotoMetry : photometric value.
- Resol : resolution
- Saturation : saturation value.
- WhiteBalance : white valance value.
- Zoom : zoom ratio.

### 1.2.8 SENSOR\_CAPA

The structure which could read the whole Capability information of the image sensor



```
Public struct SENSOR_CAPA
{
    uint bpp;
    stBRIGHTNESS Brightness;
    bool bSupportAF;
    bool bSupportFlash;
    uint ColorSpace;
    stCONTRAST Contrast;
    uint Effect;
    stEXPOSURE Exposure;
    uint FlipMirror;
    uint MaxResol;
    uint PhotoMetry;
    stSATURATION Saturation;
    uint WhiteBalance;
    stZOOM_CAPA ZoomCapa;
};
```

- bpp : the image data format that is output from the Camera Module.
- Brightness : value of brightness.
- bSupportAF : whether or not auto focus supported.
- bSupportFlash : whether or not flash light supported.
- ColorSpace : color space value.
- Contrast : contrast value.
- Effect : digital effect
- Exposure : exposure value.
- FlipMirror : screen reverse mode value.
- MaxResol : The maximum resolution supported by the module.
- PhotoMetry : photometric value.
- Saturation : saturation value.
- WhiteBalance : white valance value.
- ZoomCapa : zoom function information supported by the module.

### 1.3 Delegates

### 1.3.1 CAMERACALLBACK

Delegate function which for receiving the status of the image sensor form AADF.

Register delegate function through using SetCallback function, and CAMERAMSG, which received from parameter, indicates the status of the sensor.

Public delegate void **CAMERACALLBACK**(CAMERAMSGcamMsg);

## 1.4 Methods

### 1.4.1 Open

Open camera device and assign necessary system resource.

**RESULT** Open();

#### Parameters

*None*

#### Return Values

RESULT\_OK will be returned, if camera device open normally.

### 1.4.2 Close

Remove the assigned system resource and close the camera device.

**RESULT** Close();

#### Parameters

*None*

#### Return Values

RESULT\_OK will be returned if close the camera device properly.

### 1.4.3 IsOpen

Check whether camera device is open

**BOOL** IsOpen();

#### Parameters

*None*

#### Return Values

TRUE: camera device in the state of open.

FALSE: camera device in the state of close.

### 1.4.4 SetCallback

The delegate function which register for receiving the situation of sensor.

Using CAMERACALLBACK(CAMERAMSG camMsg) function to register the Callback function.

```
RESULT SetCallback(  
    CAMERACALLBACK pFunc,  
);
```

**Parameters**

*pFunc*

delegate function which for receiving the situation of sensor.

**Return Values**

RESULT\_OK will be returned if called function properly.

**1.4.5 SetHandle**

Setting the handle of Preview windows.

```
RESULT SetHandle(  
    IntPtr pPicboxhWnd  
);
```

**Parameters**

*pPicboxhWnd*

windows handle of application program that will drawing the preview image.

**Return Values**

RESULT\_OK will be returned if called function properly.

**1.4.6 StartPreview**

Drawing the image data, which inputted from camera module, to the preview windows.

```
RESULT StartPreview();
```

**Parameters**

*None*

**Return Values**

RESULT\_OK will be returned if performed the preview successfully.

#### 1.4.7 StopPreview

Stop Preview

```
RESULTStopPreview ();
```

##### Parameters

*None*

##### Return Values

RESULT\_OK will be returned if stop the preview properly.

#### 1.4.8 Capture

Saving the image data, which inputted form camera module, as file

```
RESULTCapture (  
    CAPTURE_SETUP_INFOCaptureParas  
);
```

##### Parameters

*CaptureParas*

Transferthe information of the camera module settings which in order to capture images, and image form which will be saved will transmitted as ref from.

##### Return Values

RESULT\_OK will be returned if performed the image capture successfully.

#### 1.4.9 GetInfo

Get the current setting values of the camera module.

```
RESULTGetInfo (  
    ref SENSOR_SETUP_INFOCamInfo  
);
```

##### Parameters

*CamInfo*

the ref form value of region where will store the current setting values.

##### Return Values

RESULT\_OK will be returned, if save the setting values into CamInfo properly.

#### 1.4.10 GetSensorCapa

Get the capability which camera module supported.

```
RESULT GetInfo (
    ref SENSOR_CAPA CamCapa
);
```

##### Parameters

*CamCapa*

theref form value of region where will save the capability of the camera module.

##### Return Values

RESULT\_OK will be returned if save the camera module capability to the CamCapa properly.

##### Notes

In order to change the setting values of the camera, firstly should know the current support values of the module through using this function, and set the value on the base of those values.

#### 1.4.11 SetAutoFocus

Automatically perform the focus adjustment.

```
RESULT SetAutoFocus (
    ref SENSOR_SETUP_INFO CamParas
);
```

##### Parameters

*CamParas*

The setting value of Camera device.

##### Return Values

RESULT\_OK will be returned if performed successfully.

#### 1.4.12 SetAutoFocusEx

Executing and setting Auto Focus mode.

```
CAM_RESULT SetAutoFocusEx (
    int Mode
);
```

##### Parameters

*Mode*

Auto Focus mode

1 : continuous mode (Executing auto focusing continuously)

2 : single mode (executing auto focusing once)

**Return Values**

If performed successfully, CAM\_RESULT\_SUCCESS will be returned.

If Mode value is not 1 or 2, CAM\_RESULT\_INVALID\_ARGS will be returned.

**1.4.13 SetBrightness**

Change brightness value of the camera module.

```
RESULT SetBrightness (  
    ref SENSOR_SETUP_INFO CamParas  
);
```

**Parameters**

*CamParas*

The Setting value of Camera device.

**Return Values**

RESULT\_OK will be returned if changed the brightness value properly.

**1.4.14 SetContrast**

Change the contrast values of the camera module.

```
RESULT SetContrast (  
    ref SENSOR_SETUP_INFO CamParas  
);
```

**Parameters**

*CamParas*

The Setting value of Camera device.

**Return Values**

RESULT\_OK will be returned if changed the contrast value properly.

**1.4.15 SetEffect**

Apply the designated digital filter effect to the images.

```
RESULT SetEffect (  

```

```
ref SENSOR_SETUP_INFO CamParas  
);
```

**Parameters**

*CamParas*

The setting value of Camera device.

**Return Values**

RESULT\_OK will be returned if applied the digital filter effect properly.

**1.4.16 SetFlipMirror**

Apply Flip/Mirror effect to image.

```
RESULTSetFlipMirror (  
    ref SENSOR_SETUP_INFO CamParas  
);
```

**Parameters**

*CamParas*

The Setting value of Camera device.

**Return Values**

RESULT\_OK will be returned if performed properly.

**1.4.17 SetSaturation**

Change saturation value of the camera module.

```
RESULTSetSaturation (  
    ref SENSOR_SETUP_INFO CamParas  
);
```

**Parameters**

*CamParas*

The Setting value of Camera device.

**Return Values**

RESULT\_OK will be returned if changed saturation properly.

**1.4.18 SetWhiteBalance**

Change the white balance value that inputted to camera module.



```
RESULTSetWhiteBalance (  
    ref SENSOR_SETUP_INFO CamParas  
);
```

**Parameters**

*CamParas*

The Setting value of Camera device.

**Return Values**

RESULT\_OK will be returned if changed the White balance successfully.

**1.4.19 EnableFlash**

Turn On/Off the flash light.

```
RESULTEnableFlash (  
    boolbEnable  
);
```

**Parameters**

*bEnable*

True:On, False:Off

**Return Values**

RESULT\_OK will be returned if turned on/off the flash light successfully.

## 2 C/C++ API Reference

### 2.1 Enumerations

#### 2.1.1 CAM\_RESULT

The result of a call to functions.

- **CAM\_RESULT\_SUCCESS**  
Function executed successfully.
- **CAM\_RESULT\_INVALID\_ARGS**  
Invalid parameter.
- **CAM\_RESULT\_OUTOFMEMORY**  
Failed to assign resource.
- **CAM\_RESULT\_UNSUPPORTED**  
No support command currently.
- **CAM\_RESULT\_ALREADY\_OPENED**  
Cam device has opened.
- **CAM\_RESULT\_ALREADY\_CAM\_SELECTED**  
Cam device was already selected.
- **CAM\_RESULT\_SCAN\_SELECTED**  
2D Barcode device was already selected.  
(Camera and 2D Barcode device can not be used at the same time)
- **CAM\_RESULT\_NOT\_OPENED**  
Call function without Open.
- **CAM\_RESULT\_FAILURE**  
Failed to perform function.
- **CAM\_RESULT\_INVALID\_DEVICE**  
Not installed Cam device.

#### 2.1.2 CAM\_BPP

- **RawRGB8Bits**
- **RGB565**
- **YCbCr422**

### **2.1.3 CAM\_COL\_SP**

- Raw
- RGB
- YCbCr

### **2.1.4 CAMERAMSG**

- CAMERA\_INITIALIZED
- CAPTURE
- COMPLETE
- PREVIEW\_DONE
- START\_TO\_SAVE

### **2.1.5 CAM\_RESOL**

- CIF
- QQVGA
- QVGA
- QXGA
- SVGA
- SXGA
- UXGA
- VGA
- XGA

### **2.1.6 CONT\_OPTIONS**

- CONT\_0
- CONT\_N1
- CONT\_N2
- CONT\_N3
- CONT\_N4
- CONT\_N5
- CONT\_P1
- CONT\_P2
- CONT\_P3

- CONT\_P4
- CONT\_P5

#### **2.1.7 EFFECT**

- EF\_BLUE
- EF\_GREEN
- EF\_MONO
- EF\_NEGATIVE
- EF\_RED
- EF\_SEPIA
- EF\_VIOLET
- NO\_EFFECT

#### **2.1.8 EV\_OPTIONS**

- EV\_0
- EV\_N0D3
- EV\_N0D7
- EV\_N1D0
- EV\_N1D3
- EV\_N1D7
- EV\_P0D3
- EV\_P0D7
- EV\_P1D0
- EV\_P1D3
- EV\_P1D7

#### **2.1.9 FLIP\_MIRROR**

- FLIP
- FLIP\_AND\_MIRROR
- MIRR
- MIRR\_FLIP\_UNSUPPORTED
- NO\_MIRROR\_FLIP

#### **2.1.10 IMG\_ENCODE\_FORMAT**

- **IMG\_ENCODE\_BMP**
- **IMG\_ENCODE\_EXIF**
- **IMG\_ENCODE\_GIF**
- **IMG\_ENCODE\_ICON**
- **IMG\_ENCODE\_JPG**
- **IMG\_ENCODE\_TIFF**

#### **2.1.11 PM\_OPTIONS**

- **PM\_NORMAL**
- **PM\_SPOT**

#### **2.1.12 SAT\_OPTIONS**

- **SAT\_025X**
- **SAT\_05X**
- **SAT\_075X**
- **SAT\_125X**
- **SAT\_15X**
- **SAT\_175X**
- **SAT\_1X**
- **SAT\_2X**
- **SAT\_0X**

#### **2.1.13 SEN\_CTRL\_CODE**

- **SEN\_CTRL\_AUTOFOCUS**
- **SEN\_CTRL\_BRIGHTNESS**
- **SEN\_CTRL\_CONTRAST**
- **SEN\_CTRL\_EFFECT**
- **SEN\_CTRL\_EXPOSURE**
- **SEN\_CTRL\_PHOTOMETRY**
- **SEN\_CTRL\_PLIP\_MIRROR**
- **SEN\_CTRL\_SATURATION**
- **SEN\_CTRL\_WHITE\_BALANCE**

#### **2.1.14 WHITE\_BAL**

- WB\_AUTO
- WB\_CLOUDY
- WB\_FLUORESCENT
- WB\_INCANDESCENT
- WB\_SUNNY
- WB\_UNSUPPORTED

## 2.2 Structures

### 2.2.1 stBRIGHTNESS

Brightness Information structure

```
typedef struct
{
    intBrightDefault
    signedintBrightMax;
    signedintBrightMin;
    charBrightSteps
}stBRIGHTNESS;
```

### 2.2.2 stCONTRAST

Contrast information structure.

```
typedef struct
{
    intContrastDefault;
    signedintContrastMax;
    signedintContrastMin;
    charContrastSteps
}stCONTRAST;
```

### 2.2.3 stEXPOSURE

Exposure information structure

```
typedef struct
{
    intExposureDefault;
    signedintExposureMax;
    signedintExposureMin;
    charExposureSteps
}stEXPOSURE;
```

#### 2.2.4 stSATURATION

Saturation information structure.

```
typedef struct
{
    int SaturationDefault;
    signed int SaturationMax;
    signed int SaturationMin;
    char SaturationSteps
}stSATURATION;
```

#### 2.2.5 stZOOM\_CAPA

The structure of information of zoom function.

```
typedef struct
{
    bool bZoomSupport;
    unsigned int MaxZoom;
    int ZoomStep;
}stZOOM_CAPA;
```

#### 2.2.6 CAPTURE\_SETUP\_INFO

The structure of the necessary information while capture image.

```
typedef struct
{
    IMG_ENCODE_FORMAT ImgFormat
    int nJpegQuality;
    SENSOR_SETUP_INFO SensorSetup;
    TCHAR pSavePath[MAX_PATH]
}CAPTURE_SETUP_INFO;
```

#### 2.2.7 SENSOR\_SETUP\_INFO

Constructor which for changing the setting of the image sensor

```
typedef struct
{
```



```

    BOOLbAutoFocus;
    BOOLbEnableAF;
    CAM_BPPbpp;
    int Brightness;
    BOOLbSupportFlash;
    CAM_COL_SPColorSpace;
    CONT_OPTIONS Contrast;
    EFFECTEffect;
    EV_OPTION Exposure;
    FLIP_MIRRORFlipMirror;
    PM_OPTIONSPhotoMetry;
    CAM_RESOLResol;
    SAT_OPTIONS Saturation;
    WHITE_BALWhiteBalance;
    int Zoom;
}SENSOR_SETUP_INFO;

```

### 2.2.8 SENSOR\_CAPA

The structure which could read the whole Capability information of the image sensor.

```

typedefstruct
{
    DWORDbpp;
    stBRIGHTNESS Brightness;
    BOOLbSupportAF;
    BOOLbSupportFlash;
    DWORDColorSpace;
    stCONTRAST Contrast;
    DWORD Effect;
    stEXPOSURE Exposure;
    DWORDFlipMirror;
    DWORDMaxResol;
    DWORDPhotoMetry;
    stSATURATION Saturation;
    DWORDWhiteBalance;
    stZOOM_CAPAZoomCapa;
}SENSOR_CAPA;

```

## **2.3 Constants**

### **2.3.1 CAM\_MGS\_OFFSET**

Message offset value

### **2.3.2 WM\_MSG\_CAMAPI**

Camera message

### **2.3.3 WM\_CAM\_DRV\_CAMERA\_INITIALIZED**

indicates that Image sensor's initialization operation has completed.

### **2.3.4 WM\_CAM\_DRV\_CAPTURE\_DONE**

indicates that image processing has completed.

### **2.3.5 WM\_CAM\_DRV\_CAPTURE\_START\_TO\_SAVE**

indicates that the starting to save images which has captured.

### **2.3.6 WM\_CAM\_DRV\_CAPTURE\_COMPLETE\_SAVE**

indicates that has stored images.

### **2.3.7 WM\_CAM\_DRV\_PREVIEW\_DONE**

indicates that the starting of preview.

## 2.4 Methods

### 2.4.1 CamOpen

Open camera device and assign necessary system resource.

```
CAM_RESULTCamOpen();
```

#### Parameters

*None*

#### Return Values

CAM\_RESULT\_SUCCESS will be returned, if camera device open normally.

### 2.4.2 CamClose

Remove the assigned system resource and close the camera device.

```
CAM_RESULTCamClose();
```

#### Parameters

*None*

#### Return Values

CAM\_RESULT\_SUCCESS will be returned if close the camera device properly.

### 2.4.3 CamIsOpen

If performed the CamOpen properly, then camera will be open status.

```
BOOLCamIsOpen ();
```

#### Parameters

*None*

#### Return Values

TRUE: camera device in the state of open.

FALSE: camera device in the state of close.

### 2.4.4 CamSetHandle

Register the windows handle(hPrevWnd) that application program which will be drawing windows handle(hWnd) and preview image, that will receive windows message from

system driver of the camera.

```
CAM_RESULT CamSetHandle(  
    HANDLE hwnd,  
    HANDLE picboxHwnd  
);
```

**Parameters**

*hwnd*

application window handle that will receive window messages.

*picboxHwnd*

preview window handle that will draw preview image

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if close the camera device properly.

**2.4.5 CamStartPreview**

Drawing the image data, which is inputted from camera module, to the preview windows.

```
CAM_RESULT CamStartPreview();
```

**Parameters**

*None*

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if performed the preview successfully.

**2.4.6 CamStopPreview**

Stop Preview

```
CAM_RESULT CamStopPreview ();
```

**Parameters**

*None*

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if stop the preview properly.

**2.4.7 CamCapture**

saving the image data, which is inputted from camera module, as file

```
CAM_RESULTCamCapture (
    CAPTURE_SETUP_INFO*CamCaptureParas
);
```

#### Parameters

*CamCaptureParas*

transfer the information of the camera module settings which in order to capture images, and image form which will be saved will be transmitted as ref from.

#### Return Values

CAM\_RESULT\_SUCCESS will be returned if performed the image capture successfully.

### 2.4.8 CamGetInfo

Get the current setting values of the camera module.

```
CAM_RESULTCamGetInfo (
    SENSOR_SETUP_INFO*CamInfo
);
```

#### Parameters

*CamInfo*

The setting value of current Camera device..

#### Return Values

CAM\_RESULT\_SUCCESS will be returned, if save the setting values into CamInfo properly.

### 2.4.9 CamGetSensorCapa

Get the capability which camera module supported.

```
CAM_RESULTCamGetInfo (
    SENSOR_CAPA*CamCapa
);
```

#### Parameters

*CamCapa*

Function supported by Camera device.

#### Return Values

CAM\_RESULT\_SUCCESS will be returned if save the camera module capability to the

CamCapa properly.

#### Notes

In order to change the setting values of the camera, firstly should know the current support values of the module through using this function, and set the value on the base of those values.

#### 2.4.10 CamSetAutoFocus

Automatically perform the focus adjustment.

```
CAM_RESULTCamSetAutoFocus (
    SENSOR_SETUP_INFO*CamParas
);
```

#### Parameters

*CamParas*

Setting value of Camera device.

#### Return Values

CAM\_RESULT\_SUCCESS will be returned if performed successfully.

#### 2.4.11 CamSetAutoFocusEx

Setting and executing Auto Focus mode.

```
CAM_RESULTCamSetAutoFocusEx (
    intnMode
);
```

#### Parameters

*nMode*

Auto Focus mode

1 : continuous mode (executing auto focusing continuously)

2 : single mode (executing auto focusing once.)

#### Return Values

If performed successfully, CAM\_RESULT\_SUCCESS will be returned.

If nMode value is not 1 or 2, CAM\_RESULT\_INVALID\_ARGS will be returned.

#### 2.4.12 CamSetBrightness

Change brightness value of the camera module.

```
CAM_RESULTCamSetBrightness (
```

```
SENSOR_SETUP_INFO*CamParas  
);
```

**Parameters**

*CamParas*

Setting value of Camera device.

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if changed the brightness value properly.

**2.4.13 CamSetContrast**

Change the contrast values of the camera module.

```
CAM_RESULTCamSetContrast (  
    SENSOR_SETUP_INFO*CamParas  
);
```

**Parameters**

*CamParas*

Setting value of Camera device.

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if changed the contrast value properly.

**2.4.14 CamSetEffect**

Apply the designated digital filter effect to the images.

```
CAM_RESULTCamSetEffect (  
    SENSOR_SETUP_INFO*CamParas  
);
```

**Parameters**

*CamParas*

Setting value of Camera device.

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if applied the digital filter effect properly.

**2.4.15 CamSetFlipMirror**

Apply Flip/Mirror effect to image.

```
CAM_RESULTCamSetFlipMirror (  
    SENSOR_SETUP_INFO*CamParas  
);
```

**Parameters**

*CamParas*

Setting value of Camera device.

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if applied Flip/Mirror effects properly.

**2.4.16 CamSetSaturation**

Change saturation value of the camera module.

```
CAM_RESULTCamSetSaturation (  
    SENSOR_SETUP_INFO*CamParas  
);
```

**Parameters**

*CamParas*

Setting value of Camera device.

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if changed saturation properly.

**2.4.17 CamSetWhiteBalance**

Change the white balance value that inputted to camera module.

```
CAM_RESULTCamSetWhiteBalance (  
    SENSOR_SETUP_INFO*CamParas  
);
```

**Parameters**

*CamParas*

Setting value of Camera device..

**Return Values**

CAM\_RESULT\_SUCCESS will be returned if changed the White balance successfully.



#### 2.4.18 CamEnableFlash

Turn On/Off the flash light.

```
CAM_RESULT CamEnableFlash (  
    bool bEnable  
);
```

##### Parameters

*bEnable*

True:On, False:Off

##### Return Values

CAM\_RESULT\_SUCCESS will be returned if turned on/off the flash light successfully.